Log: Week 8

This week I have:

- Implemented basic/ placeholder genetic operators
- These allow the algorithm to accurately learn/ approximate straight line routes from $A \to B$

Below you can see the population after 1 generation:
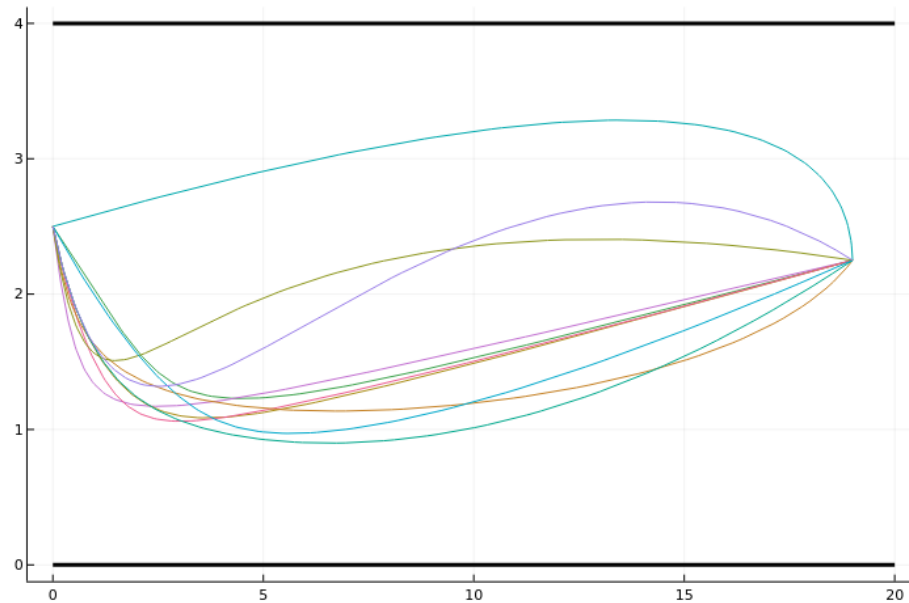

Figure 1: 1 Generation

And the most and least fit solutions after 4 generations

Whilst in isolation this is not particularly impressive, in fact is under performs a simple Pythagorean distance calculation, it can be generalised and extended to avoid obstacles and to not intersect with other routes.

I went on to implement road space obstacles in the form of circles. I have implemented the abstract type of `Obstacle` of which `Circle` is a member allowing me to extend my program.

By calculating infeasible route sections as the distance between 2 intersects between an individual and a obstacle, I can penalise solutions where such a distance is non-zero.

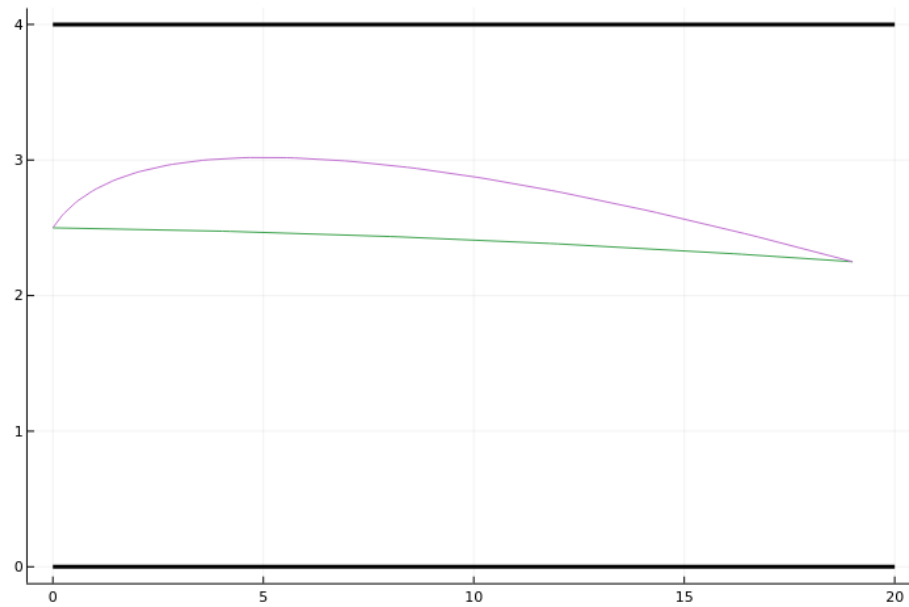The results can be seen in Figure 3. These results are seen after just a single

Figure 2: 4 Generations min max

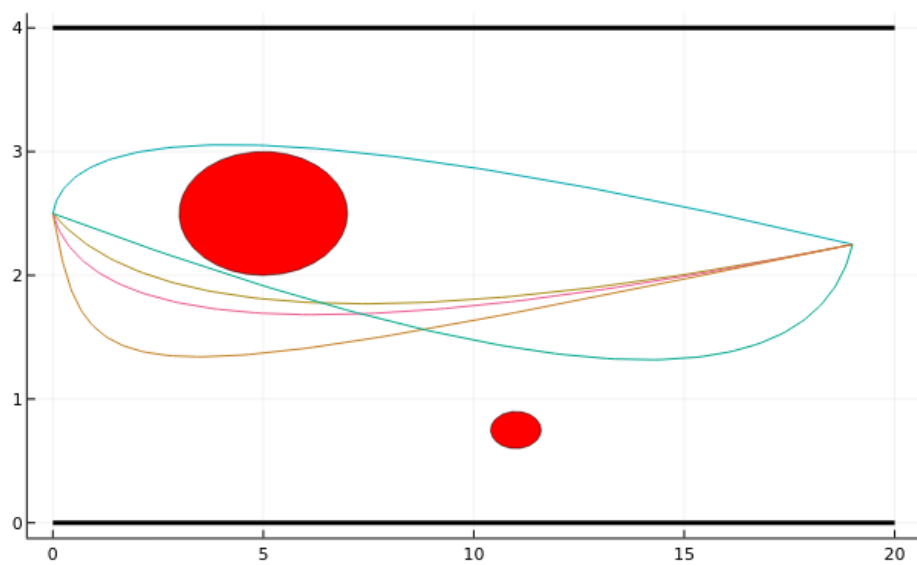generation with the best soluton having approximately 8% lower fitness than the pythagorian route.

Figure 3: Obstacle avoidance